

**REMARKS**

After receipt of the Advisory Action dated August 20, 2004, in which the Examiner indicated that the Amendment After Final would not be entered, Applicants' representative contacted the Examiner to ask why the change to the phrase "may access" would require another search. During the telephone call, the Examiner indicated that, in the meantime, she had found other art, U.S. Pat. No. 6,272,612 to Bordaz et al. ("Bordaz"), that allegedly anticipated all the claims of the application. The Examiner suggested that Applicants submit a Request for Continued Examination (RCE) and amend the claims in view of Bordaz. She did not withdraw the finality of the Office Action or provide a new Office Action detailing her new grounds of rejection. Applicants filed an RCE on September 23, 2004, requesting that that the Amendment filed June 22, 2004, should be entered. Applicants also requested a three-month suspension of action on the RCE form and authorized the charge to a deposit account in the appropriate amount of \$130.

**Request to Withdraw Premature Office Action**

When Applicants received an Office Action mailed December 1, 2004, Applicants' representatives contacted the Examiner to request that the Office Action be withdrawn as premature in light of Applicants' request for a three-month suspension of action. The Examiner verified that the three-month suspension of action was properly requested and indicated that the Office Action mailed December 1, 2004, would be withdrawn.

**Pending claims**

Applicants cancel claims 9, 12-14, and 16-20 without prejudice or disclaimer. Applicants amend claim 1 to add an "A" before "computer system," to follow standard

practice. Applicants add new claims 21-31. Claims 21-26 and 29-31, in general, correspond to claims 8-13 and 16-18, respectively, as first presented in the amendment filed January 6, 2004. Claims 27 and 28 correspond, in general, to original claims 14 and 15, but are directed toward a method and not a scheme.

**Patentability over Bordaz**

Applicants submit that independent claims 1, 21, 27, 29, 30, and 31, as well as those claims that depend directly or indirectly from them, are patentable over Bordaz for at least the following reasons. Bordaz fails to disclose at least, "the database system stores at least a first assignment of a first predetermined profile to the first memory portion and at least a second assignment of a second predetermined profile to the second memory portion, wherein the first and second profiles are unique and refer to the first and second application systems, respectively" as recited by claim 1.

Bordaz discloses an improved method of allocating physical memory "in a multiprocessor data processing system, more specifically a process for allocating a memory having non-uniform access." Col. 1, lines 6-9. The allocation occurs in response to a page fault detected when a software application requests a virtual memory address but the address correspondence table mapping the virtual memory space to global physical memory does not have that virtual address in it. Col. 8, lines 28-31.

"For the applications running in this environment [“data processing system whose operating system is “UNIX” or a similar type”], the virtual address space can be divided into different types, including: text or program text (executable code); initialized data; modified data; ‘stacks’; ‘heap’, that is dynamic allocation (tables, etc.); shared memory; and shared libraries.” Col. 6, lines 35-45. “[W]hen a given application is running, its

various components are partitioned into virtual memory segments (text, data, stack, shared memory, files, etc.)" within the divided virtual memory space discussed above. Col. 7, lines 8-12. These segments of virtual memory are then allocated physical locations in the global physical memory according to rules. Col. 7, lines 12-15. As an example, when a virtual memory address in the "text" virtual memory portion of the divided virtual memory space is sought by an application, if that "text type" virtual memory address is not in the address correspondence table, "there is the transmission of an exception, which translates into a page fault detected by the handler H." Col. 7, lines 25-32.

When the handler detects a page fault, the allocation method of Bordaz may be used. "[E]ach application is linked to a set of predefined allocation rules. To do this, it is necessary to associate each application . . . with a particular profile. This association is carried out under the control of the operating system, or 'OS,' which stores it." Col. 7, lines 63 – Col. 8, line 1.

The profile is a record of the various types of memory that the application uses, e.g., text, data, etc, and the corresponding memory allocation rules chosen to optimize performance. Col. 8, lines 3-5, 14-18, and Fig. 6a. A handler searches the profile associated with the application that is requesting the virtual memory address to determine the appropriate memory allocation rule for the type of virtual memory address not found in the address correspondence table. Col. 8, lines 31-38. When it matches the type of virtual memory address to the type of memory in the profile, it uses the corresponding memory allocation rule assigned to that type of memory. Col. 8, lines 32-55. It then uses that rule, for example, "P\_LOCAL," which tells it to assign physical

memory space from the local main memory attached to the processor running the application. Col. 8, lines 65-67.

In the example of Fig. 6a of Bordaz, six types of memory are used by the application, and each is assigned a specific rule, "it being understood that this is not necessarily a set of disjointed rules. In other words, for example, the rules R2 and R3 could be identical, even if memory types tyM2 and tyM3 themselves are different." Col. 8, lines 5-14.

Elsewhere, in discussing another preferred embodiment, Bordaz makes "clear that the term 'specific' does not necessarily imply that the allocation policy associated with a given range . . . is different from that associated with one or more other virtual ranges." Col. 4, lines 59-64.

Thus it is apparent that Bordaz does not disclose "a database system . . . wherein the first and second profiles are unique and refer to the first and second application systems, respectively," as recited by claim 1. Moreover, unique profiles would not assist Bordaz in accomplishing its disclosed goal "to best optimize the physical memory allocations as a function of the actual needs of the applications, or more precisely the particular profiles of the applications," col. 9, lines 12-14, if two applications used the same memory types. See, e.g., col. 9, lines 27-29 ("Moreover, a 'daughter' application can 'inherit' memory allocation rules associated with the 'mother' application that created it."). As Bordaz fails to disclose or suggest each and every limitation of claim 1, Applicants submit that claim 1 is patentable over Bordaz and should be allowed.

Application No. 09/989,070  
Attorney Docket 7781.0040-00

As claims 2-7 each depend directly or indirectly from claim 1, Applicants submit that they are also patentable over Bordaz for at least the reasons stated with respect to claim 1 and should be allowed.

Each of new independent claims 21, 27, 29, 30, and 31 also recites a limitation regarding unique profiles. For at least that reason, Applicants submit that they, and the claims that depend directly or indirectly from them, are also patentable over Bordaz, and should be allowed.

As a final note, Applicants amended the specification to remove references to specific claim numerals. No new matter has been added.

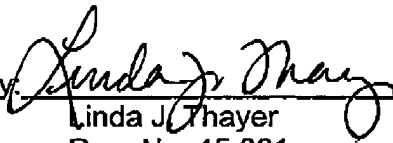
In view of the foregoing amendments and remarks, Applicants respectfully request the reconsideration and reexamination of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,  
GARRETT & DUNNER, L.L.P.

Dated: December 22, 2004

By:   
Linda J. Thayer  
Reg. No. 45,681